



Progetto:

**EvenTrento**

Titolo del documento:

**Analisi e Progettazione**

Autore:

**Togni Roberto**

Document Info:

Doc. Name	D2-EvenTrentoProgettazione	Doc. Number	D2 V0.1
Description	Documento di descrizione dell'implementazione		

**Dipartimento di Ingegneria e Scienza dell'Informazione**

**Contents**

<b>1</b>	<b>Scopo del documento</b>	<b>i</b>
<b>2</b>	<b>Requisiti Funzionali</b>	<b>ii</b>
2.1	FR1 e FR2: Login e Registrazione . . . . .	ii
2.2	FR4: Visualizzazione Eventi . . . . .	iii
2.3	FR5, FR6, FR16 e FR17: Feedback, Condivisione, Iscrizione e salvataggio Eventi . . . . .	iv
2.4	FR8 e FR10: Creazione ed Aggiornamento Eventi . . . . .	iv
2.5	FR12-13: Aggiunta e Modifica Luogo . . . . .	vi
<b>3</b>	<b>Sequence and Activity Diagram</b>	<b>viii</b>
<b>4</b>	<b>Analisi dei Componenti</b>	<b>xviii</b>
4.1	Definizione dei Componenti . . . . .	xviii
4.2	Diagramma dei Componenti . . . . .	xxi
<b>5</b>	<b>Diagramma delle Classi</b>	<b>xxii</b>
5.1	Diagrammi delle Classi Parziali . . . . .	xxii
5.2	Diagramma delle Classi Complessivo . . . . .	xxv
<b>6</b>	<b>Dal Class Diagrams alle APIs</b>	<b>xxvi</b>

## **1 Scopo del documento**

Il seguente documento riporta la specifica dei requisiti funzionali del sistema tramite un linguaggio semi-formale. Si tratta dunque di un approfondimento (nonché di una formalizzazione) di quanto riportato in linguaggio naturale all'interno del D1. Il linguaggio utilizzato per la formalizzazione dei requisiti è UML (Unified Modeling Language), declinato in Use Case Diagrams (UCDs), Component Diagrams, Sequence Diagrams, e Class Diagrams.

## 2 Requisiti Funzionali

Di seguito sono riportati i functional requirements (FR) del sistema sia in linguaggio naturale che tramite Use Case Diagrams (UCDs). La notazione è coerente con quella utilizzata all'interno del documento D1.

### 2.1 FR1 e FR2: Login e Registrazione

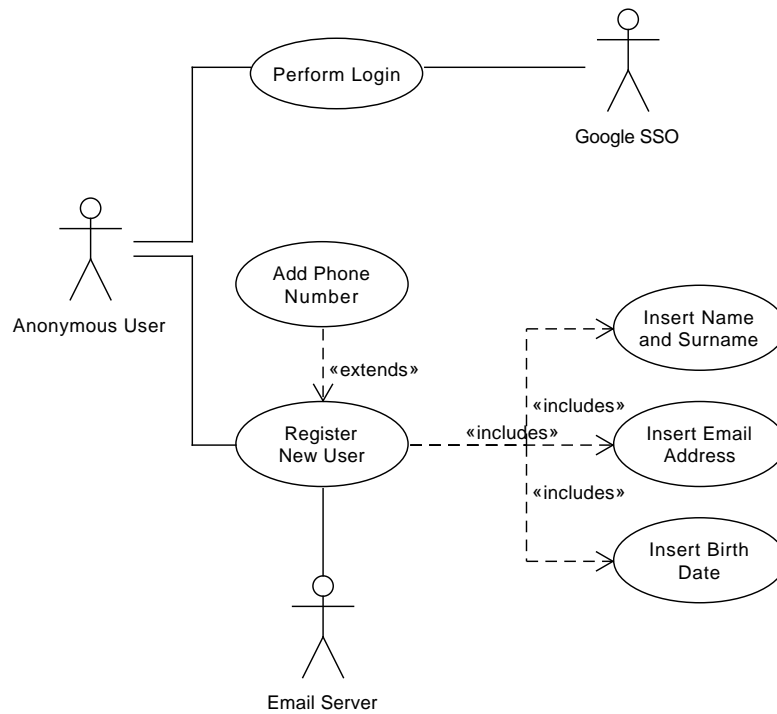


Figure 1: UCD relativo a FR1 e FR2.

#### Use Case FR1: Login

##### Riassunto

Lo use case in questione descrive come un utente può effettuare il login al sistema.

##### Descrizione

- L'utente visualizza una pagina con un form
- Se l'utente inserisce le credenziali (vedi Estensioni) corrette (vedi Eccezioni) e preme sul pulsante "Login", si apre la schermata principale dell'applicazione.
- Qualora l'utente selezioni il logo di Google, il processo di autenticazione verrà gestito da Google SSO

#### Use Case FR2: Registrazione

##### Riassunto

Lo use case in questione descrive come un utente può registrarsi al sistema.

### Descrizione

- L'utente visualizza una pagina dalla quale inserire nome, cognome, indirizzo mail (vedi Eccezioni)
- Il sistema invia una mail all'indirizzo fornito dall'utente contenente un link e una password temporanea
- L'utente deve confermare la registrazione tramite il link di cui sopra (vedi Eccezioni). Può quindi accedere al sistema tramite la password temporanea

### Eccezioni

- Qualora le credenziali inserite non siano corrette, l'applicazione restituisce un messaggio di errore
- Qualora l'utente non apra il link contenuto nella mail automatica, la registrazione non viene finalizzata

### Estensioni

- La password contenuta nella mail generata automaticamente dal sistema a seguito della registrazione dev'essere modificata a seguito del primo login

## 2.2 FR4: Visualizzazione Eventi

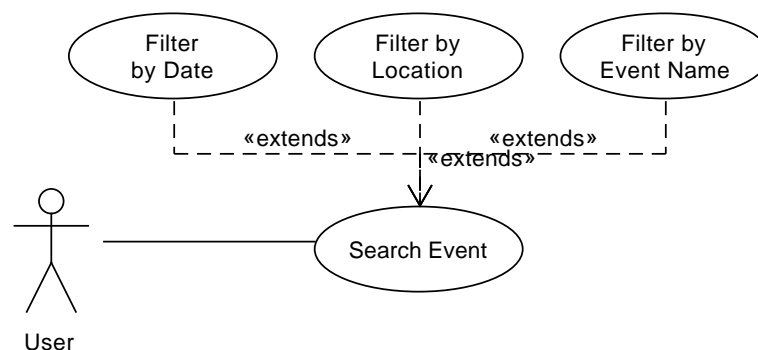


Figure 2: UCD relativo al FR4.

### Use Case FR4: Visualizzazione Eventi

#### Riassunto

Lo use case in questione descrive come un utente può filtrare gli eventi presenti all'interno del sistema.

#### Descrizione

- La ricerca di eventi può essere effettuata da qualsiasi utente, anche non loggato
- L'utente può scegliere se esplorare la mappa integrata nella schermata iniziale, o se visualizzare gli eventi in forma di lista

- Qualora l'utente decida di visualizzare la lista degli eventi, viene messo a disposizione un servizio di filtering basato su 3 possibili criteri: data, luogo e nome dell'evento

### 2.3 FR5, FR6, FR16 e FR17: Feedback, Condivisione, Iscrizione e salvataggio Eventi

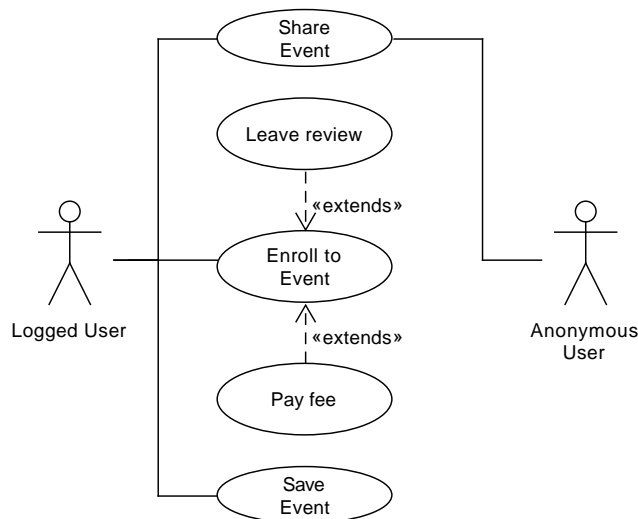


Figure 3: UCD relativo a FR5, FR6, FR16 e FR17.

#### Use Cases FR5, FR6, FR16, FR17

##### Riassunto

Lo use case in questione descrive come un utente può interagire con gli eventi offerti dal sistema. In particolare, descrive le attività di salvataggio, di iscrizione, di condivisione, nonché di valutazione della qualità di un evento.

##### Descrizione

- Oltre a poter visualizzare gli eventi (come mostrato nello UCD precedente), qualsiasi utente può condividerli
- Previo login, qualsiasi tipologia di utente può salvare e/o iscriversi ad un evento. L'iscrizione comporta l'eventuale pagamento del biglietto
- Qualsiasi utente iscritto ad un evento può, dopo il suo verificarsi, lasciare una review

### 2.4 FR8 e FR10: Creazione ed Aggiornamento Eventi

#### Use Case FR8: Creazione Eventi

##### Riassunto

Lo use case in questione descrive come un utente appartenente alla categoria "Organizer" oppure "Owner" può creare nuovi eventi.

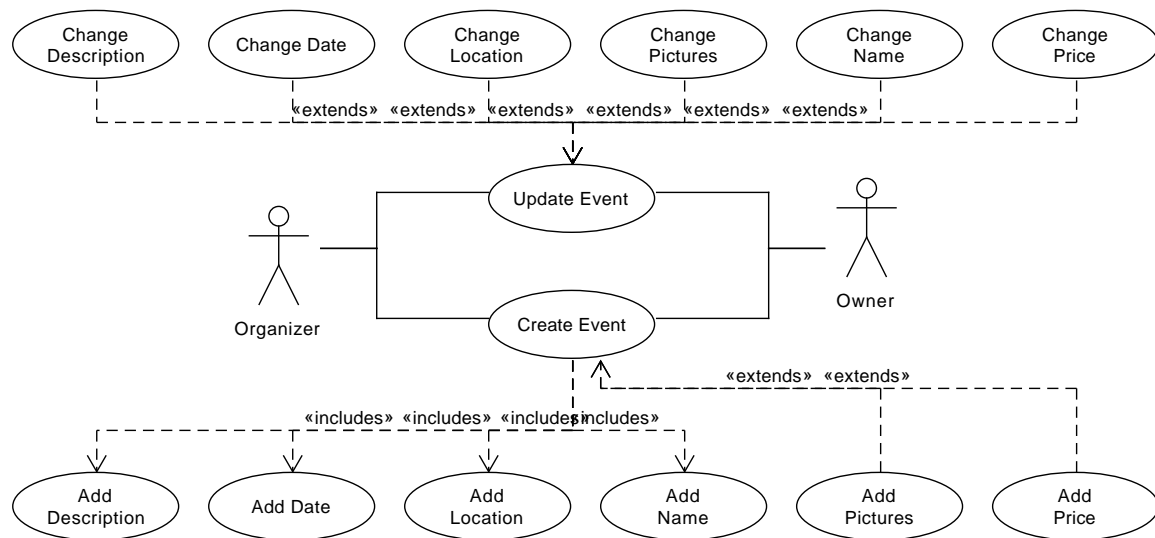


Figure 4: UCD relativo a FR8 e FR10.

## Descrizione

- Qualora l'utente appartenga alla categoria "Owner" o a quella "Organizer" e sia loggato (vedi Eccezioni), dalla pagina del profilo personale è visibile un pulsante "New event"
- La pressione del pulsante rimanda alla pagina di creazione eventi. Per la creazione di un nuovo evento è necessario l'inserimento di un nome, di una data, di una location e di una descrizione (vedi Eccezioni)
- L'aggiunta di immagini è facoltativa, così come lo è il prezzo

## Use case FR10: Modifica Eventi

### Riassunto

Lo use case in questione descrive come un utente appartenente alla categoria "Organizer" oppure "Owner" può modificare eventi precedentemente creati.

### Descrizione

- Qualora un utente appartenente alle categorie "Owner" o "Organizer" sia loggato e abbia precedentemente creato un evento (vedi Eccezioni), dalla propria pagina personale può raggiungere l'evento in questione tramite il pulsante "My events"
- La pressione del pulsante rimanda ad una lista degli eventi creati. Selezionandone uno è possibile modificare uno qualsiasi dei vari campi entro una settimana dall'evento (vedi Eccezioni e Estensioni)

### Eccezioni

- Qualora l'utente non sia loggato, non esiste alcun profilo personale

- Qualora l'utente appartenga alla categoria "User", la pagina del profilo personale è priva del pulsante per la creazione di eventi
- Se durante la creazione di un evento non viene compilato uno dei campi obbligatori, il pulsante "Create Event" rimane inattivo
- Qualora l'utente non abbia mai creato eventi, il pulsante "My Events" non è attivo
- Qualora manchi meno di una settimana all'evento in questione, i campi non risultano modificabili

#### Estensioni

- Gli eventi passati rimangono visualizzabili tramite il pulsante "My events", tuttavia non risultano più editabili

## 2.5 FR12-13: Aggiunta e Modifica Luogo

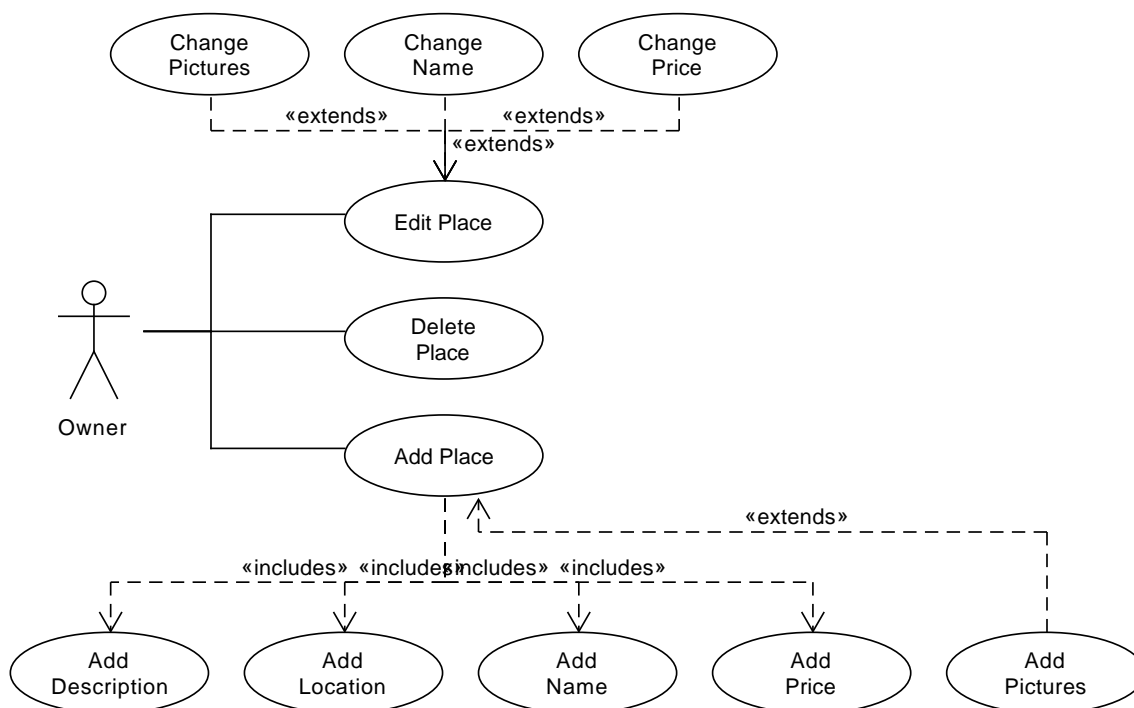


Figure 5: UCD relativo al FR12.

### Use Case FR12: Aggiunta Luogo

#### Riassunto

Lo use case in questione descrive come un utente appartenente alla categoria "Owner" può aggiungere una location.

#### Descrizione

- Qualora l'utente sia loggato con un profilo di tipo "Owner" (vedi Eccezioni), dal proprio profilo personale ha la possibilità di aggiungere un luogo tramite il pulsante



"New Place"

- La pressione del pulsante rimanda alla pagina di aggiunta luoghi. Per l'aggiunta di un nuovo luogo occorre inserire una location, un nome, una descrizione, ed un prezzo (vedi Eccezioni). Opzionalmente è possibile inserire delle foto

#### Use Case FR13: Modifica Luogo

##### Riassunto

Lo use case in questione descrive come un utente appartenente alla categoria "Owner" può modificare una location precedentemente inserita nel sistema.

##### Descrizione

- Qualora l'utente sia loggato con un profilo di tipo "Owner" e abbia precedentemente aggiunto una location, dalla propria pagina personale può raggiungere l'evento in questione tramite il pulsante "My places"
- La pressione del pulsante rimanda ad una lista degli spazi aggiunti. Selezionandone uno è possibile modificare uno qualsiasi dei campi e/o eliminare lo spazio se e solo se lo spazio non è attualmente collegato ad un evento (vedi Eccezioni)

##### Eccezioni

- Qualora l'utente non sia loggato, non esiste alcun profilo personale
- Qualora l'utente non appartenga alla categoria "Owner", la pagina del profilo personale è priva del pulsante per l'aggiunta di spazi
- Se durante l'aggiunta di uno spazio non viene compilato uno dei campi obbligatori, il pulsante "Create Place" rimane inattivo
- Qualora lo spazio sia collegato ad un evento, le modifiche sono disattivate a partire da 7 giorni prima della data prevista per l'evento

### 3 Sequence and Activity Diagram

Gli use case diagrams riportati nel capitolo precedente rappresentano una overview delle funzionalità del sistema. Nel presente capitolo tali funzionalità vengono estese tramite l'introduzione di un aspetto temporale. L'impiego di sequence diagrams consente quindi di rappresentare l'ordine cronologico con cui i vari elementi dei singoli scenari si verificano, dettagliando il modo in cui gli oggetti collaborano al fine di ottenere la piena funzionalità del sistema.

#### Login

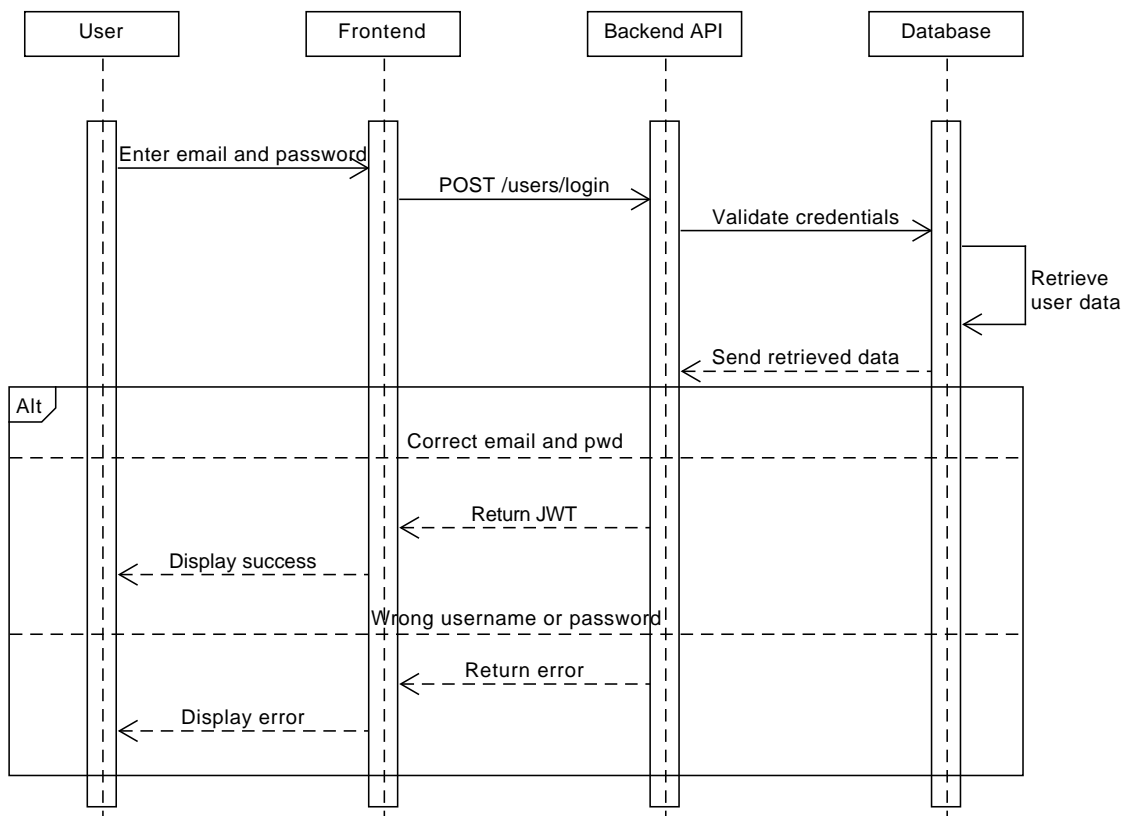


Figure 6: Sequence diagram rappresentante la procedura di login.

## Registrazione utente

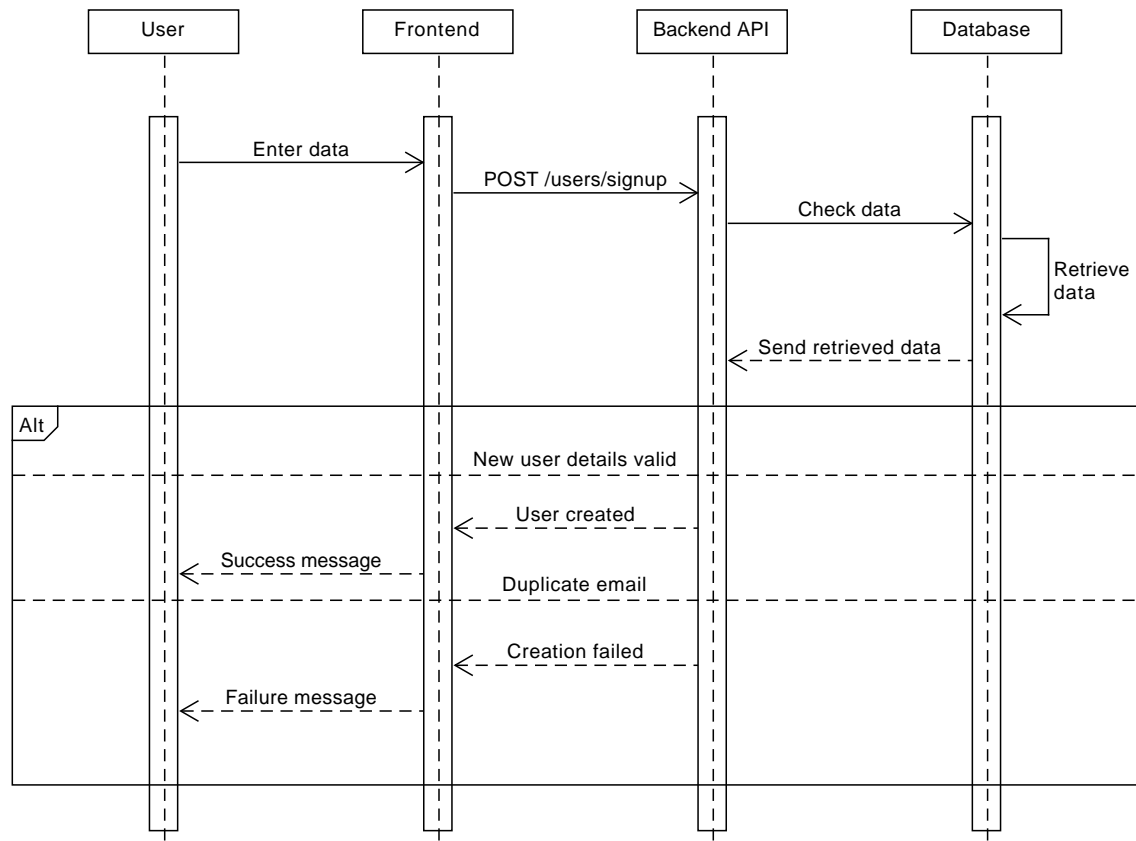


Figure 7: Sequence diagram rappresentante la procedura di registrazione.

## Browsing eventi

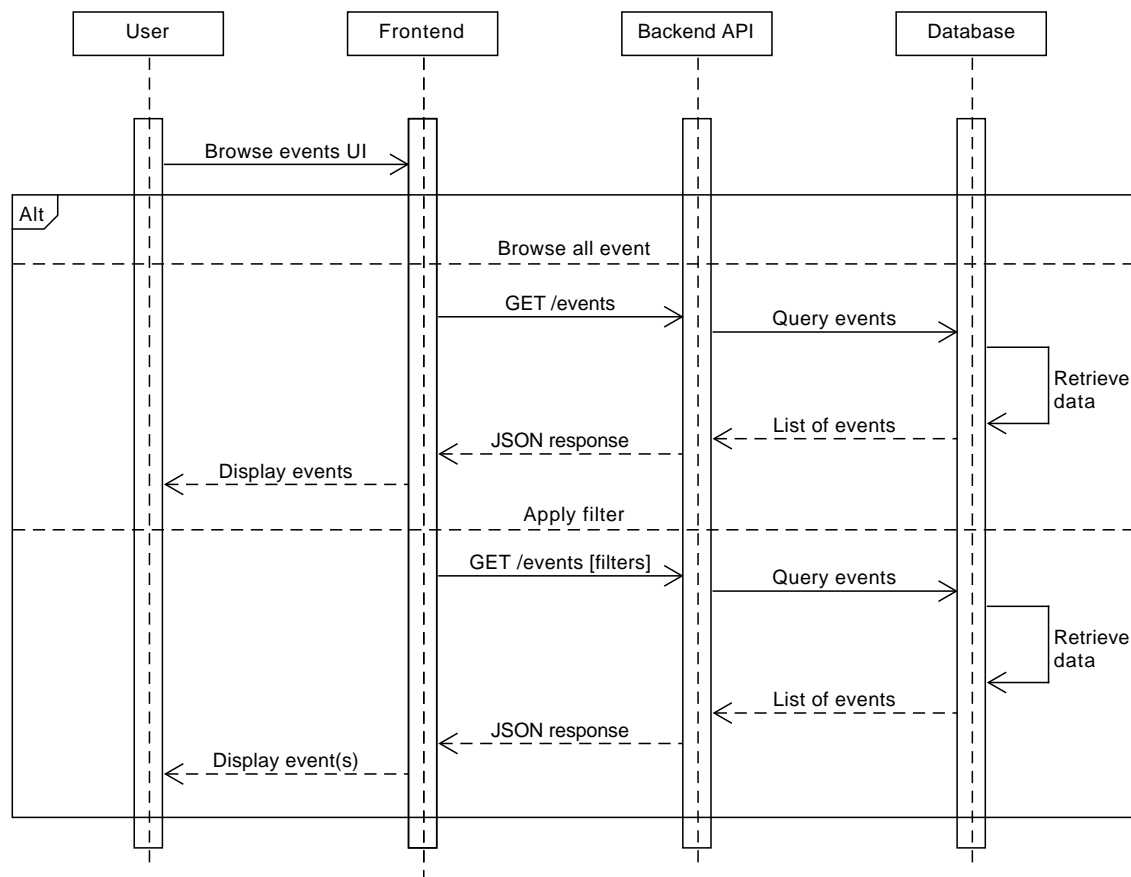


Figure 8: Sequence diagram rappresentante il browsing degli eventi.

## Iscrizione ad un evento

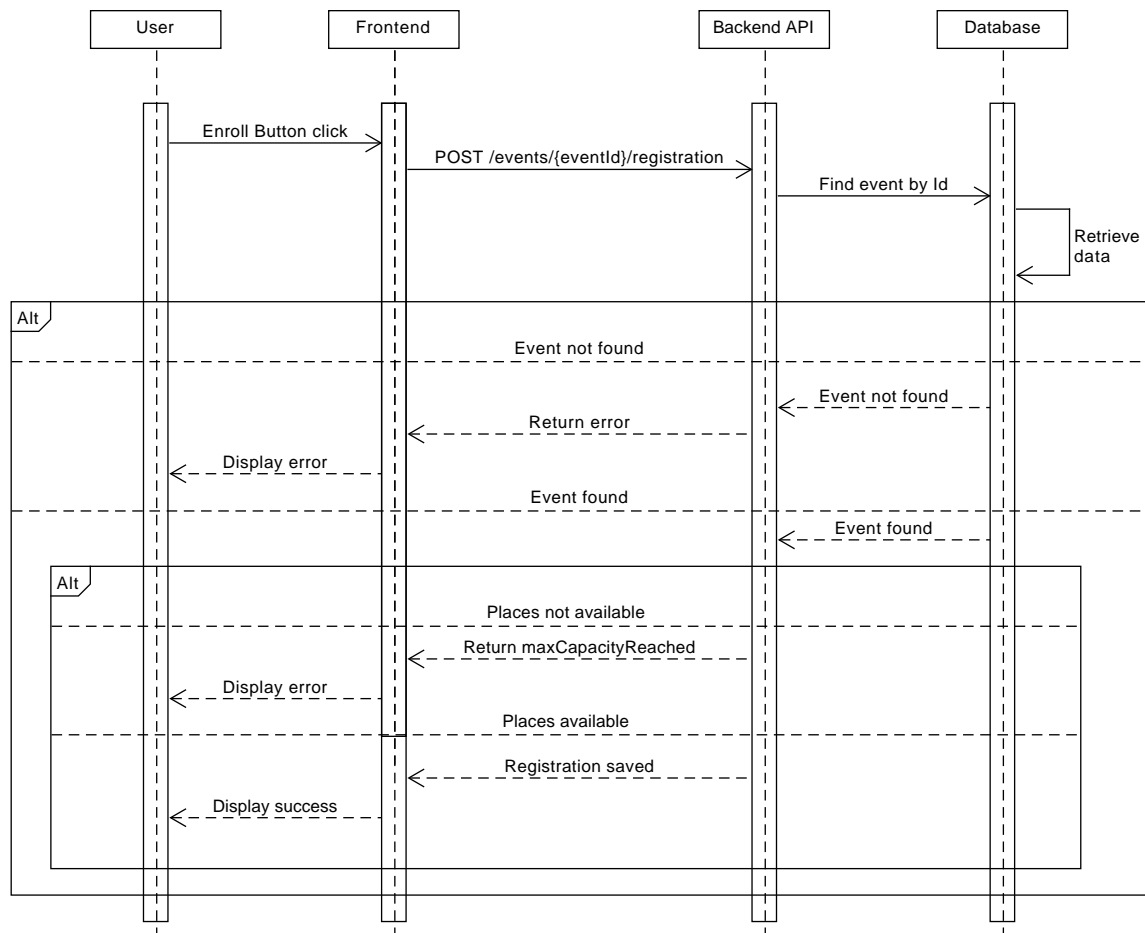


Figure 9: Sequence diagram rappresentante l'iscrizione ad un evento.

## Creazione di un evento

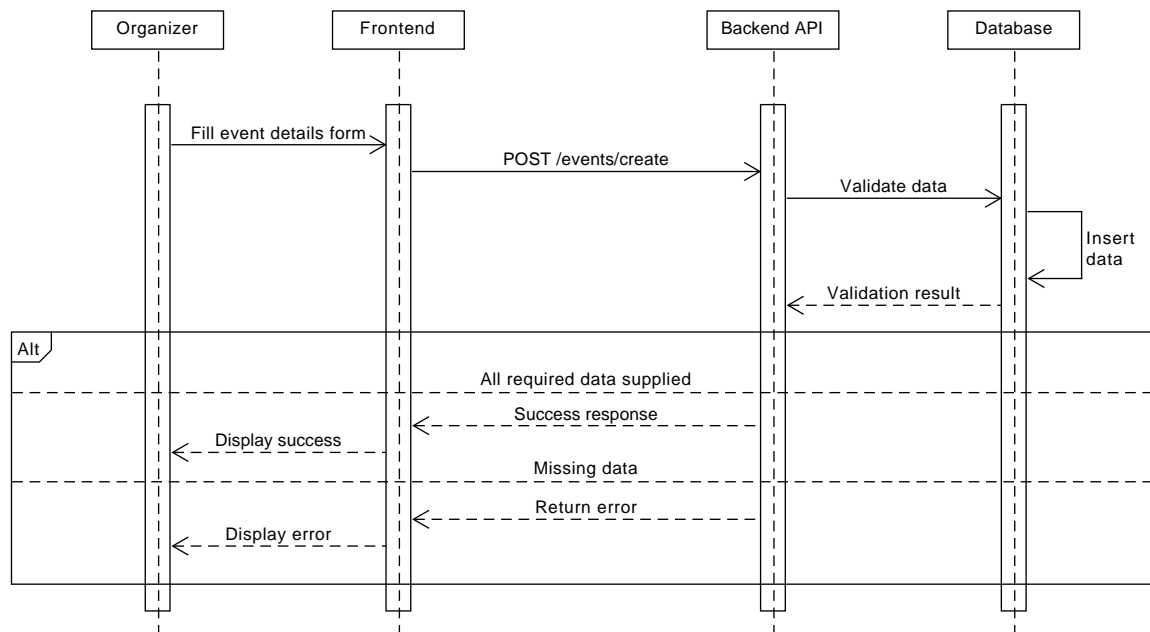


Figure 10: Sequence diagram rappresentante la creazione di un evento.

## Eliminazione di un evento

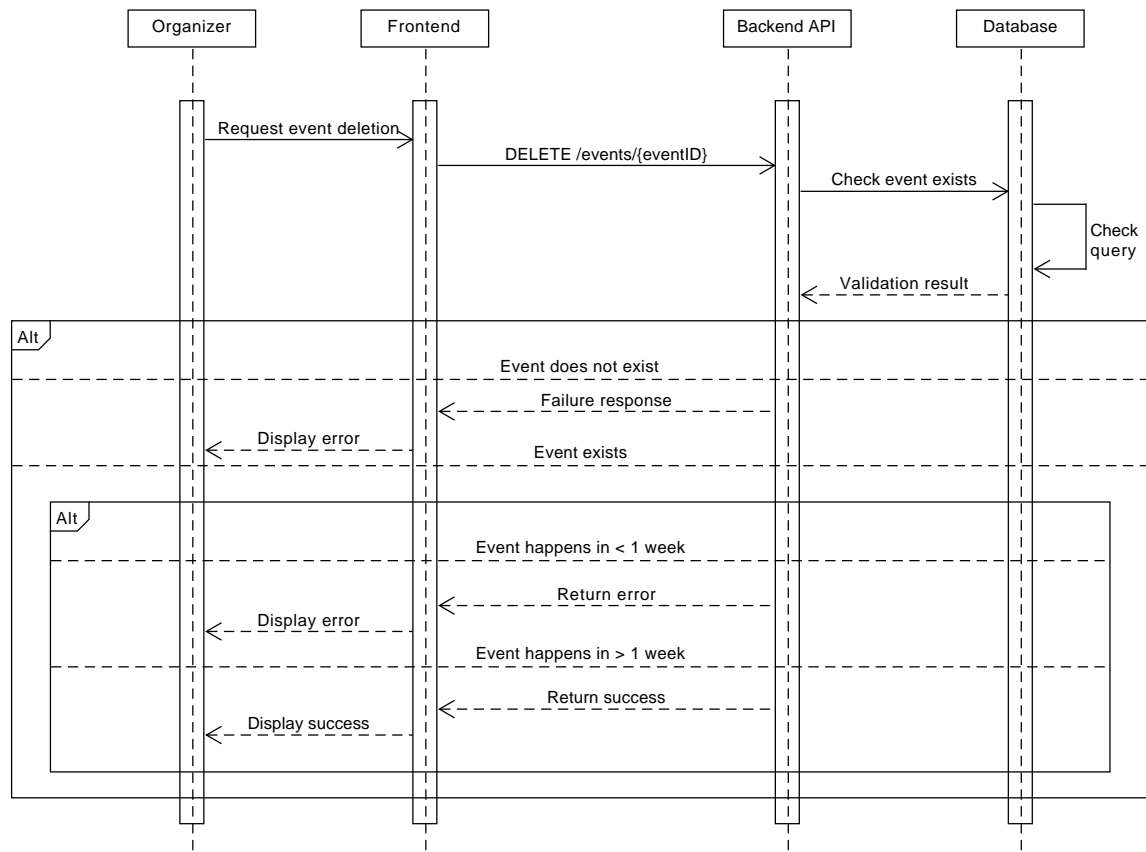


Figure 11: Sequence diagram rappresentante la rimozione di un evento creato in precedenza.

## Modifica di un evento

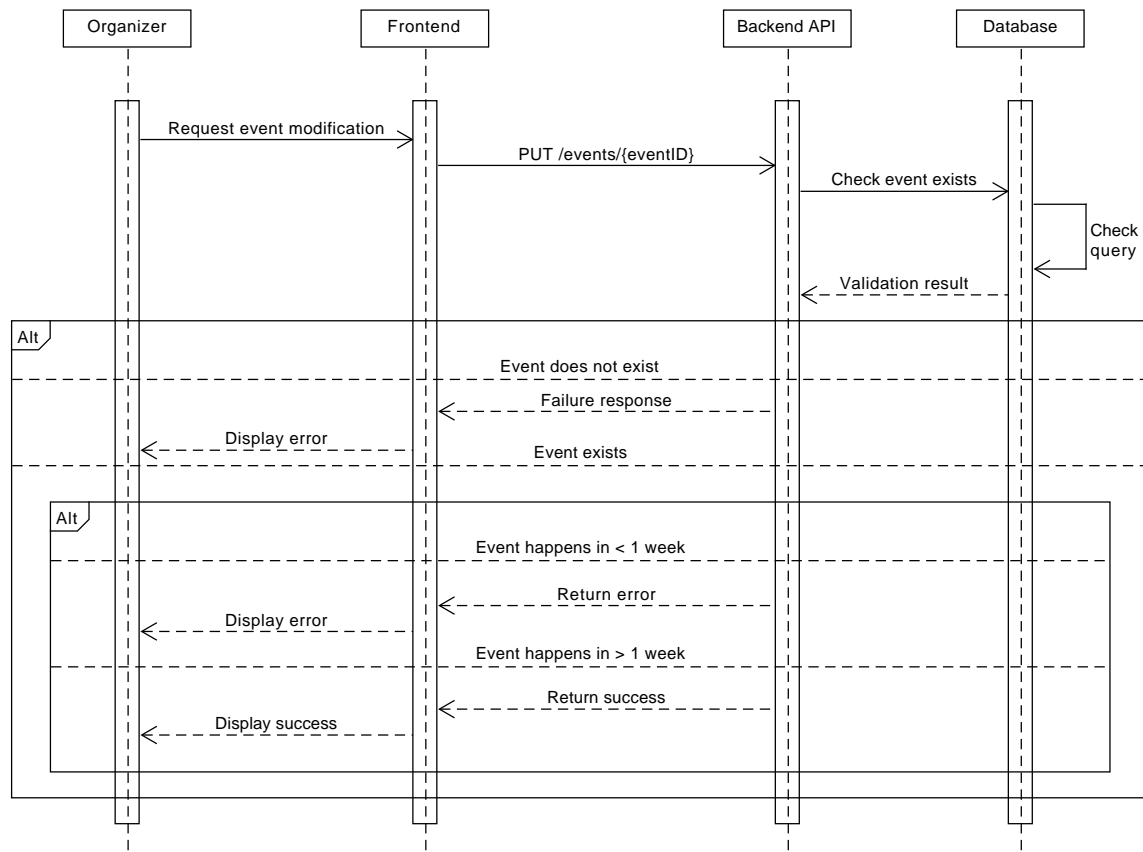


Figure 12: Sequence diagram rappresentante la modifica di un evento creato in precedenza.



## Aggiunta di un luogo

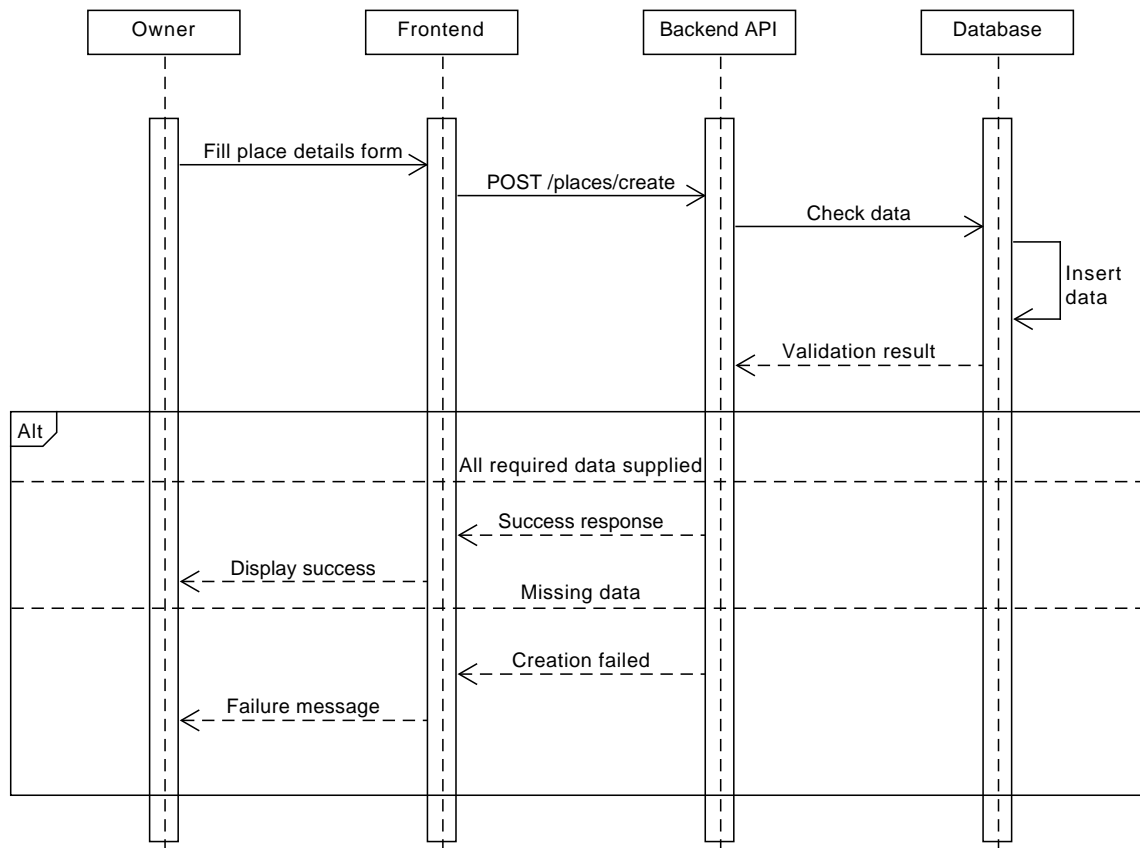


Figure 13: Sequence diagram rappresentante l'aggiunta di un luogo.

## Rimozione di un luogo

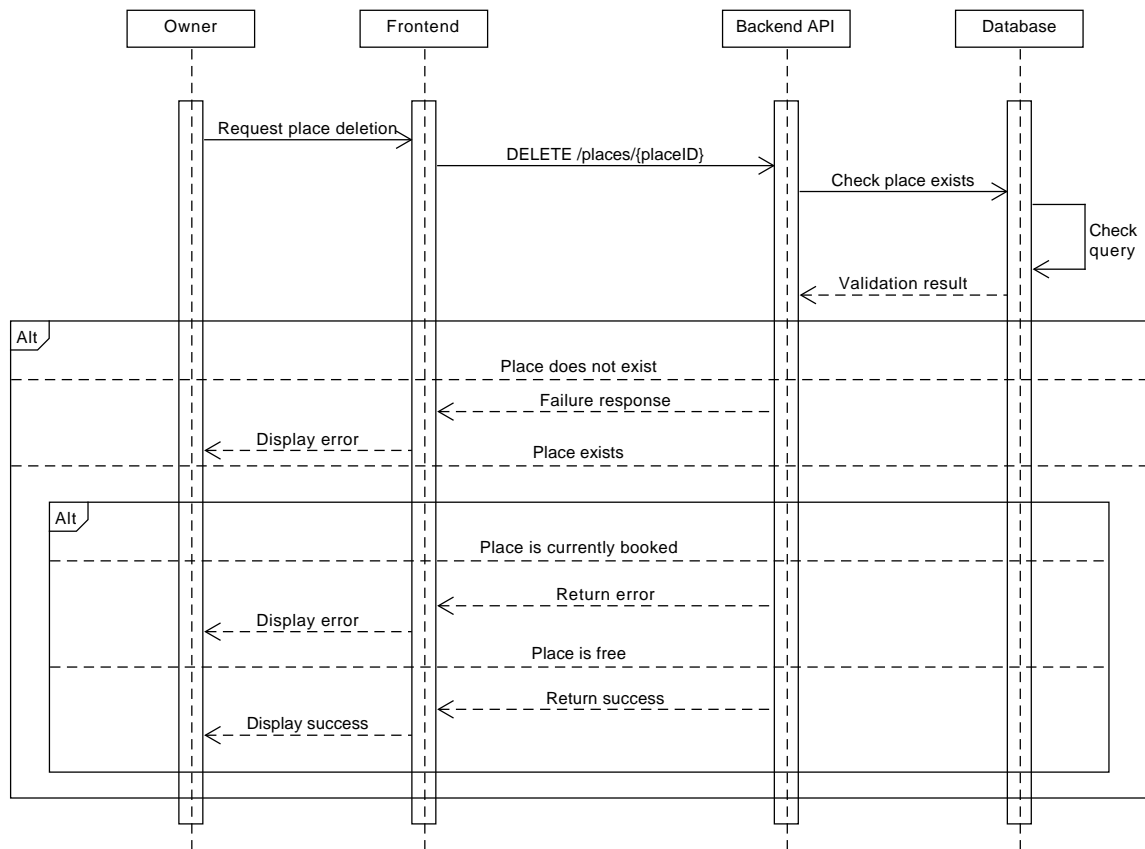


Figure 14: Sequence diagram rappresentante la rimozione di un luogo.

## Modifica di un luogo

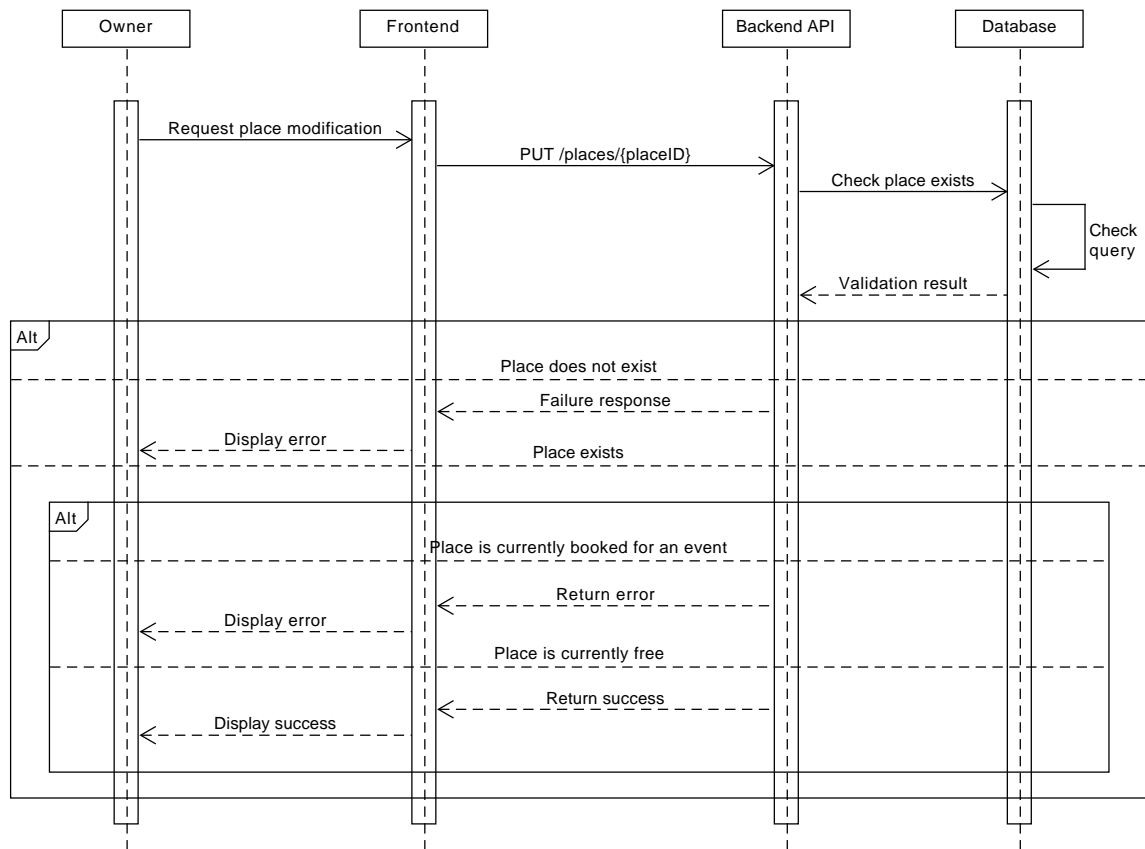


Figure 15: Sequence diagram rappresentante la modifica di un luogo.

## 4 Analisi dei Componenti

In questo capitolo vengono presentati i componenti che costituiscono l'architettura del sistema, e che sottostanno alle funzionalità definite in precedenza. L'interconnessione tra i vari componenti è rappresentata tramite l'utilizzo di un component diagram, il quale rende esplicita la presenza di interfacce sia tra i componenti stessi che con sistemi esterni.

### 4.1 Definizione dei Componenti

#### **CMP1: Gestione Registrazione**

**Descrizione** Il componente si occupa della funzionalità di registrazione di un nuovo account.

##### **Interfacce Richieste**

- Credenziali di accesso: le credenziali includono email e password

##### **Interfacce Fornite**

- Nuovo utente: una volta completata la procedura di registrazione, il nuovo profilo viene salvato all'interno del database

#### **CMP2: Gestione autenticazione**

**Descrizione** Il componente gestisce login e logout dal sistema. Interagisce con gli altri componenti per gestire la separazione tra attività che richiedono un profilo e attività aperte agli utenti non registrati.

##### **Interfacce Richieste**

- Credenziali di accesso: le credenziali includono email e password

##### **Interfacce Fornite**

- Autenticazione: garantisce che l'utente sia loggato, permettendo dunque agli altri componenti di fornire funzionalità specifiche agli utenti con un profilo di un certo tipo

#### **CMP3: Gestione database**

**Descrizione** Il componente salva i dati relativi ad utenti, eventi e luoghi all'interno del database.

##### **Interfacce Richieste**

- Credenziali utente: quando un nuovo utente si registra al sistema, i dati vengono salvati nel database
- Dati relativi ai luoghi: la componente deve avere accesso alla variazione dei dati determinata dall'update di un luogo precedentemente inserito

- Dati profilo: la componente deve avere accesso alla variazione dei dati personali causata da una modifica del profilo utente
- Dati relativi agli eventi: la componente deve avere accesso alla variazione dei dati determinata dall'update di un evento precedentemente inserito

**Interfacce Fornite**

- Dati relativi ai luoghi: la componente fornisce le informazioni relative ai luoghi presenti nel database
- Dati degli utenti: la componente fornisce le informazioni relative ai profili utenti presenti nel database
- Dati relativi agli eventi: la componente fornisce le informazioni relative agli eventi presenti nel database

**CMP4: Gestione profilo utente**

**Descrizione** Il componente gestisce le modifiche del profilo utente (e.g., cambio password)

**Interfacce Richieste**

- Autenticazione: la gestione del profilo è una funzionalità offerta solamente agli utenti che ne possiedono uno

**Interfacce Fornite**

- Modifica del profilo: permette all'utente di modificare la password

**CMP5: Gestione eventi**

**Descrizione** Il componente si occupa tutte le funzionalità legate agli eventi, consentendo agli utenti di tipo "organizer" di gestire la loro creazione, modifica e cancellazione.

**Interfacce Richieste**

- Autenticazione: la gestione degli eventi è una funzionalità offerta esclusivamente agli utenti di categoria "organizer"

**Interfacce Fornite**

- Creazione evento: crea un nuovo evento e lo inserisce nel database
- Modifica evento: aggiorna la descrizione di un evento esistente e informa il database della modifica
- Cancellazione evento: elimina dal database un evento precedentemente creato

**CMP6: Sistema di ricerca eventi**

**Descrizione** Il componente consente di effettuare la ricerca di eventi presenti nel database.

**Interfacce Richieste**

- Eventi: i dati relativi agli eventi presenti nel database
- Criteri di ricerca: i filtri da utilizzare nella ricerca degli eventi

**Interfacce Fornite**

- Eventi: gli eventi che coincidono con i filtri di ricerca utilizzati

**CMP7: Gestione iscrizione eventi**

**Descrizione** Il componente consente l'iscrizione ad un evento

**Interfacce Richieste**

- Autenticazione: solo gli utenti in possesso di un profilo possono iscriversi ad un evento
- Dati evento: le informazioni relative all'evento a cui l'utente loggato desidera iscriversi

**CMP8: Gestione mappa**

**Descrizione** Il componente mostra una mappa con la posizione degli eventi disponibili

**Interfacce Richieste**

- Eventi: i dati relativi agli eventi

**Interfacce Fornite**

- Eventi: la posizione degli eventi, sottoforma di pallini sulla mappa

**CMP9: Gestione luoghi**

**Descrizione** Il componente si occupa tutte le funzionalità legate ai luoghi, consentendo agli utenti di tipo "owner" di gestire la loro creazione, modifica ed eliminazione.

**Interfacce Richieste**

- Autenticazione: la gestione dei luoghi è una funzionalità offerta esclusivamente agli utenti di categoria "owner"

**Interfacce Fornite**

- Creazione luogo: aggiunge un nuovo luogo al database
- Modifica luogo: aggiorna la descrizione di un luogo esistente e informa il database della modifica
- Cancellazione luogo: rimuove un luogo dal database, rendendolo non più disponibile

**CMP10: Sistema di ricerca luoghi**

**Descrizione** Il componente consente di effettuare la ricerca di luoghi presenti nel database.

**Interfacce Richieste**

- Autenticazione: la ricerca di un luogo viene fatta durante la creazione o la modifica di un evento. Si tratta dunque di una funzionalità offerta solamente agli utenti di tipo "organizer"
- Luoghi: i dati relativi ai luoghi presenti nel database
- Criteri di ricerca: i filtri da utilizzare nella ricerca dei luoghi

**Interfacce Fornite**

- Luoghi: i luoghi che coincidono con i filtri di ricerca utilizzati

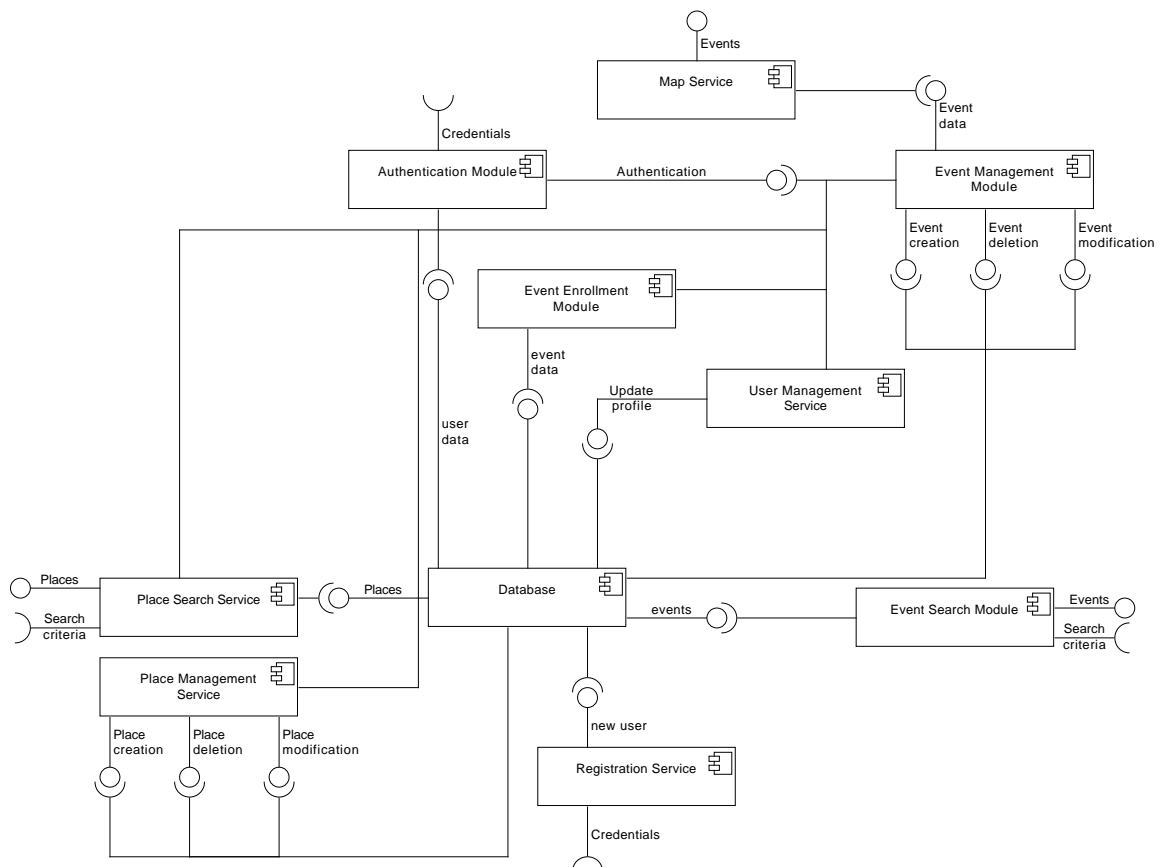
**4.2 Diagramma dei Componenti**

Figure 16: Diagramma complessivo dei componenti di sistema e delle loro interconnessioni.

## 5 Diagramma delle Classi

Nel seguente capitolo vengono presentate le classi previste nell'ambito del progetto "Even-Trento". Ogni classe rappresenta un elemento del sistema, ed è caratterizzata da un nome, da una lista di attributi e da (opzionale) una lista di metodi. Mentre le classi rappresentano i dati gestiti e/o utilizzati dalla classe, i metodi definiscono le operazioni rese disponibili dalla classe in questione. L'interazione tra le varie classi è codificata tramite le connessioni rese disponibili dalla sintassi UML.

Di seguito sono riportate le classi individuate, raggruppate per sottosistema di appartenenza.

### 5.1 Diagrammi delle Classi Parziali

#### Categorie di utente

I tre tipi di utente previsti dal sistema sono formalizzate mediante l'utilizzo di ereditarietà. In particolare, la classe **BasicUser** include tutte le caratteristiche offerte ad utenti appartenenti a tutte e tre le categorie. Ogni utente è associato a dei dati personali, e può effettuare quanto previsto dai requisiti funzionali tramite i metodi `register()`, `login()`, `logout()`, `saveEvent(eventId)`, `getSavedEvents()`, e `changePwd()`.

La classe **Organizer** estende **BasicUser** aggiungendo tutte le attività collegate alla creazione, alla modifica e alla rimozione di eventi. Analogamente, la classe **Owner** estende **BasicUser** tramite l'aggiunta delle attività relative alla gestione degli spazi.

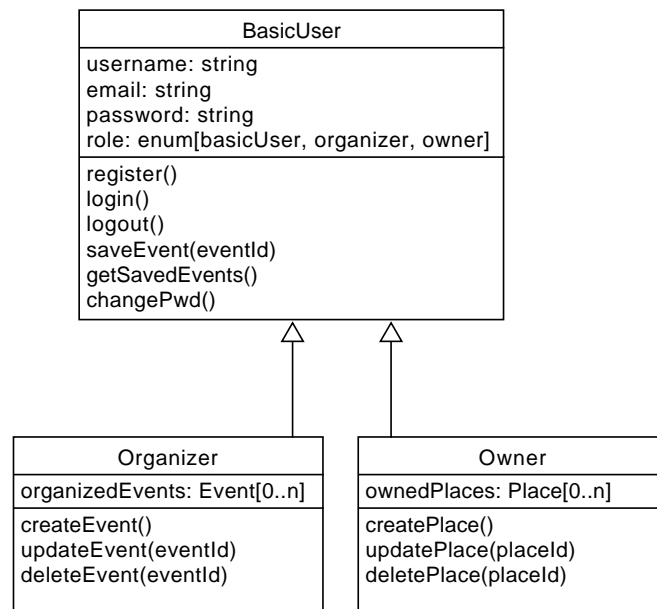


Figure 17



## Autenticazione

L'autenticazione è gestita tramite JWT tokens. Tale processo è rappresentato mediante la classe **Authentication**, come mostrato in Figura 18.

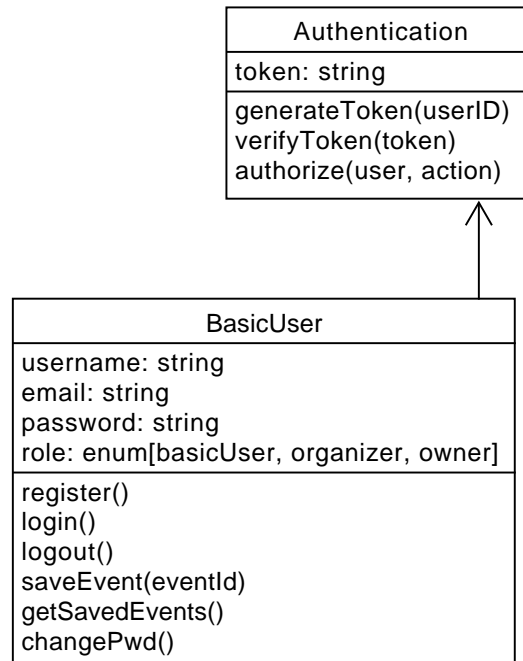


Figure 18

## Gestione eventi

Gli eventi sono rappresentati dalla classe **Event**, la quale si appoggia alle classi :

- **Place** per la scelta del luogo in cui si svolgerà l'evento
- **Date** per specificare il giorno previsto per l'evento
- **EventDescription** per fornire una descrizione dettagliata dell'evento

La ricerca di eventi e l'iscrizione ad essi sono gestite rispettivamente dalle classi **EventSearch** e **EventEnrollment**, le quali a loro volta comunicano con la classe **BasicUser**.

La classe **Event** è inoltre collegata alla classe **Organizer**, a rappresentare l'attività di creazione eventi associata a tale categoria di utente.

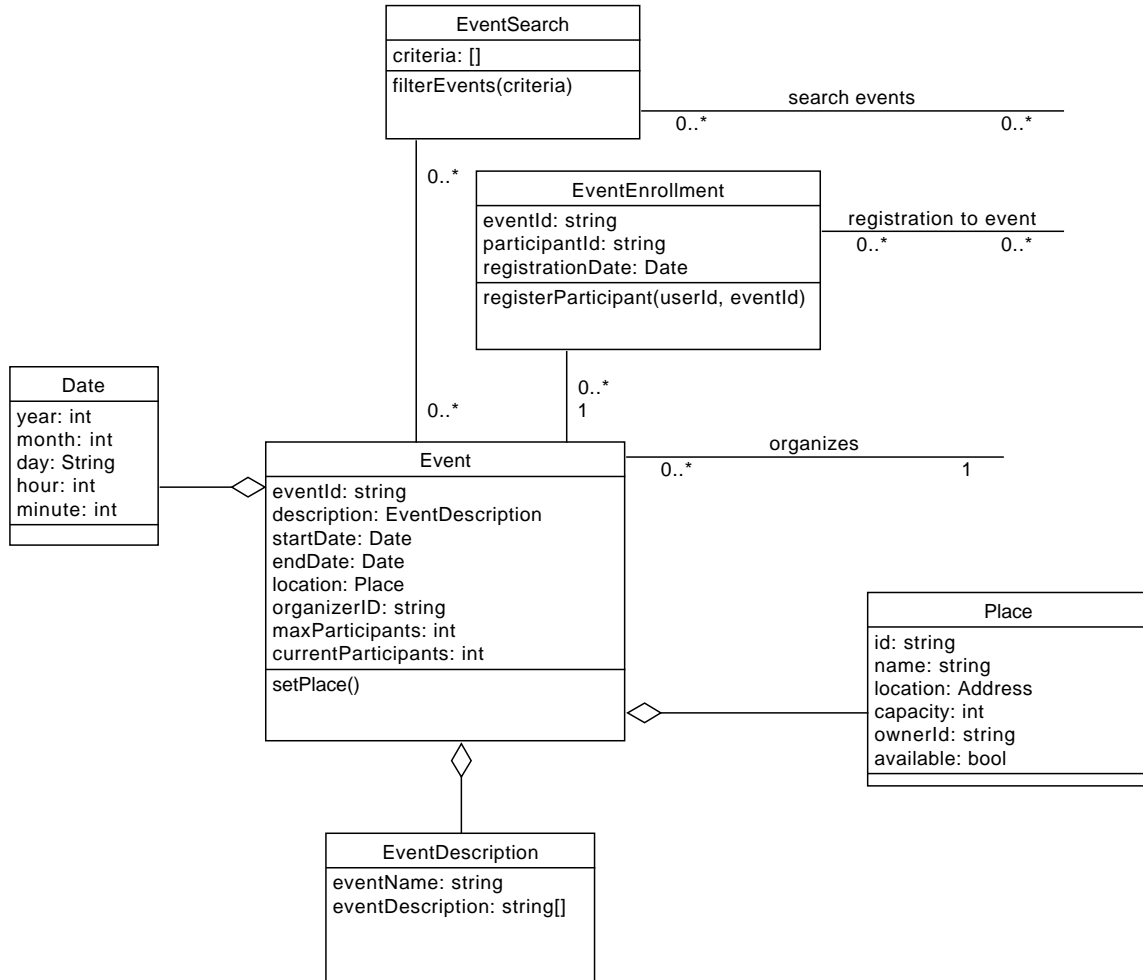


Figure 19

## Gestione degli spazi

Gli spazi sono rappresentati dalla classe **Places**, la quale si appoggia alla classe **Address** per meglio rappresentare la collocazione del locale. L'attributo **available** ha lo scopo di rappresentare la disponibilità del locale, in modo che questo non possa essere scelto come location per più eventi simultaneamente.

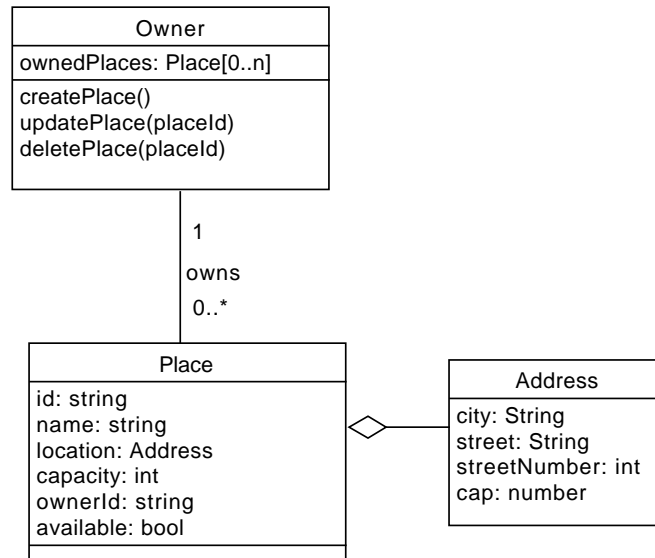


Figure 20

## 5.2 Diagramma delle Classi Complessivo

Di seguito è riportato il diagramma di tutte le classi precedentemente descritte.

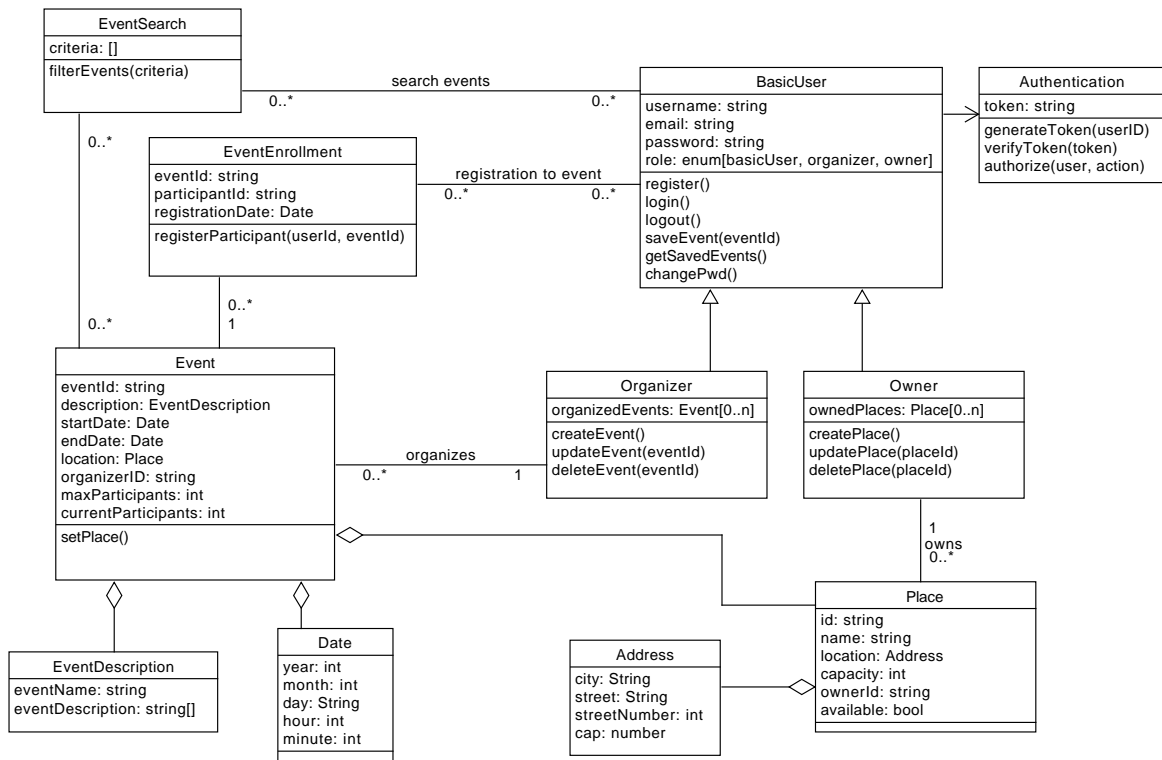


Figure 21

## 6 Dal Class Diagrams alle APIs

Lo scopo di questo capitolo consiste nel rappresentare un punto di incontro tra l'aspetto progettuale e quello implementativo del progetto. In particolare, nella seguente tabella sono riportati i vari metodi presenti all'interno del class diagram mostrato in Figura 21 e la loro controparte in termini di APIs.

Classe	Metodo	HTTP Method	URI + query params	Request	Response	Note
BasicUser	register()	POST	/api/v1/users/register	JSON email, password	JSON token, userData	Crea un nuovo utente
	login()	POST	/api/v1/users/login	JSON email, password	JSON token, userData	Restituisce il token JWT
	logout()	POST	/api/v1/users/logout	Header Authorization: Token	JSON message	Invalida il token
	getSavedEvents()	GET	/api/v1/users/savedEvents	Header Authorization: Token	JSON savedEvents	Ottiene gli eventi salvati
	saveEvent(eventId)	POST	/api/v1/users/save/eventId	Header Authorization: Token	JSON message	Salva un evento nei preferiti
	changePwd()	PUT	/api/v1/users/change-password	JSON oldPwd, newPwd	JSON message	Modifica la password
Organizer	createEvent()	POST	/api/v1/events	JSON eventData	JSON event	Crea un evento
	updateEvent(eventId)	PUT	/api/v1/events/eventId	JSON eventData	JSON event	Aggiorna un evento
	deleteEvent(eventId)	DELETE	/api/v1/events/eventId	Header Authorization: Token	JSON message	Elimina un evento
Owner	createPlace()	POST	/api/v1/places	JSON placeData	JSON place	Crea un luogo
	updatePlace(placeId)	PUT	/api/v1/places/placeId	JSON placeData	JSON place	Modifica un luogo
	deletePlace(placeId)	DELETE	/api/v1/places/placeId	Header Authorization: Token	JSON message	Elimina un luogo
Event	setPlace()	PUT	/api/v1/events/eventId/place	JSON placeId	JSON event	Assegna un luogo a un evento
	setDate()	PUT	/api/v1/events/eventId/date	JSON date	JSON event	Imposta la data di un evento
EventSearch	filterEvents(criteria)	GET	/api/v1/events?filter=criteria	Query params	JSON events	Filtra gli eventi
EventEnrollment	registerParticipant(userId, eventId)	POST	/api/v1/events/eventId/register	Header Authorization: Token	JSON message	Registra un utente a un evento
Authentication	generateToken(userId)	POST	/api/v1/auth/token	JSON userId	JSON token	Genera un JWT
	verifyToken(token)	POST	/api/v1/auth/verify	Header Authorization: Token	JSON valid: true/false	Verifica il token
	authorize(user, action)	Middleware	(interno)	Header Authorization: Token	(interno)	Middleware di autorizzazione

Table 1: Dal class diagram alle APIs.